

ARTÍCULOS DE TECNOLOGÍA &gt; FRONT END

# Simulando una API REST con json-server



viniciosneves

21 de Noviembre



En las empresas es habitual consumir APIs REST en nuestros frontends. Eventualmente, puede ser necesario desarrollar el frontend primero o en paralelo con el backend. Cuando esto sucede, tenemos que simular de alguna manera los datos enviados por el backend, es decir, decimos que estamos trabajando con mocks o "mockando" el backend. Mock no es más que el nombre que se le da a un contenido inventado y utilizado únicamente para marcar una posición. Hay varias maneras de hacer esto. Una de ellas es por json-server.

Brevemente, json-server simulará una API REST, con todos los puntos finales de un recurso: GET, POST, PUT y DELETE. Por lo tanto, nuestra interfaz consume esta API simulada, lo que permite crear toda la capa HTTP de la aplicación. Json-server es un paquete NPM que se puede instalar globalmente en ambientes que tienen instalado NodeJS. Entonces, nuestro primer paso es ejecutar la instalación:

```
npm install -g json-server
```

Una vez hecho esto, ya podemos ejecutar el comando json-server en el terminal, desde cualquier directorio:

```
+ ~ json-server
json-server [options] <source>

Options:
  -c, --config          Path to config file [default: "json-server.json"]
  -p, --port            Set port [default: 3000]
  -H, --host            Set host [default: "localhost"]
  -w, --watch           Watch file(s) [boolean]
  -r, --routes          Path to routes file
  -m, --middlewares    Paths to middleware files [array]
  -s, --static          Set static files directory
      --read-only, --ro Allow only GET requests [boolean]
      --no-cors, --nc  Disable Cross-Origin Resource Sharing [boolean]
      --no-gzip, --ng  Disable GZIP Content-Encoding [boolean]
  -S, --snapshots      Set snapshots directory [default: "."]
  -d, --delay          Add delay to responses (ms)
  -i, --id             Set database id property (e.g. _id) [default: "id"]
      --foreignKeySuffix, --fks Set foreign key suffix (e.g. _id as in post_id) [default: "Id"]
  -q, --quiet          Suppress log messages from output [boolean]
  -h, --help           Show help [boolean]
  -v, --version        Show version number [boolean]

Examples:
  json-server db.json
  json-server file.js
  json-server http://example.com/db.json

https://github.com/typicode/json-server

Missing <source> argument
```

Tenemos varias opciones disponibles para usar, pero comencemos creando un archivo JSON que representará nuestro backend. Vamos a crear dos recursos:

## proyectos,tareas

El **proyecto** es una entidad simple, que tiene un nombre y un ID incremental numérico. La tarea tiene una descripción, un ID numérico y un proyecto. Entonces podemos representarlos:

```
{
  "proyectos": [
    {
      "id": 1,
      "nombre": "Alura Tracker 3.0"
    },
    {
      "id": 2,
      "nombre": "ByteBank 2.0"
    }
  ],
  "tareas": [
    {
      "id": 1,
      "descripcion":
        "Configuración del ambiente",
      "proyecto": {
        "id": 1,
        "nombre": "Alura Tracker 3.0"
      }
    },
    {
      "descripcion": "Refactorización del vuex 4",
      "proyecto": {
        "id": 1,
        "nombre": "Alura Tracker 3.0"
      },
      "id": 2
    }
  ]
}
```

```
]
}
```

Tenga en cuenta que esta es una estructura JSON simple, guardemos este archivo como db.json. Tenemos una matriz con proyectos y otra con tareas, siguiendo la estructura definida anteriormente. Ahora podemos ejecutar el comando base:

```
json-server db.json
```

```
PS C:\Users\Barbara\Desktop\alura-trabalho\AluraGeek-lbr> json-server db.json
```

```
\{^_^}/ hi!
```

```
Loading db.json
Done
```

#### Resources

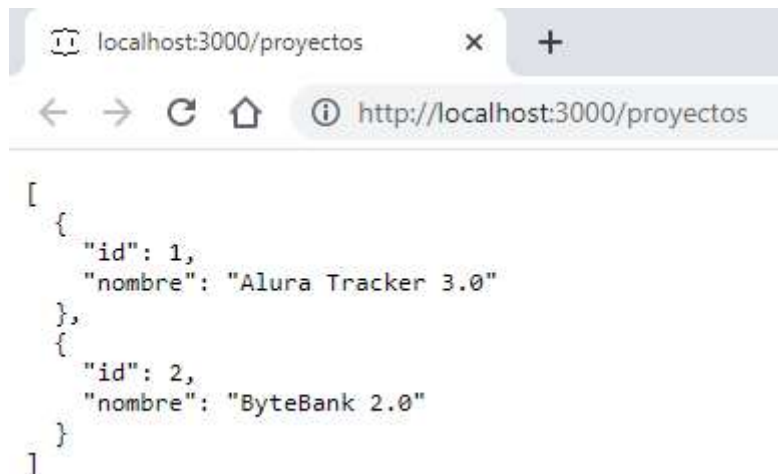
```
http://localhost:3000/proyectos
http://localhost:3000/tareas
```

#### Home

```
http://localhost:3000
```

```
Type s + enter at any time to create a snapshot of the database
```

## Accediendo a la ruta de proyectos:



The screenshot shows a web browser with the address bar set to `localhost:3000/proyectos`. The page displays a JSON array of two project objects:

```
[
  {
    "id": 1,
    "nombre": "Alura Tracker 3.0"
  },
  {
    "id": 2,
    "nombre": "ByteBank 2.0"
  }
]
```

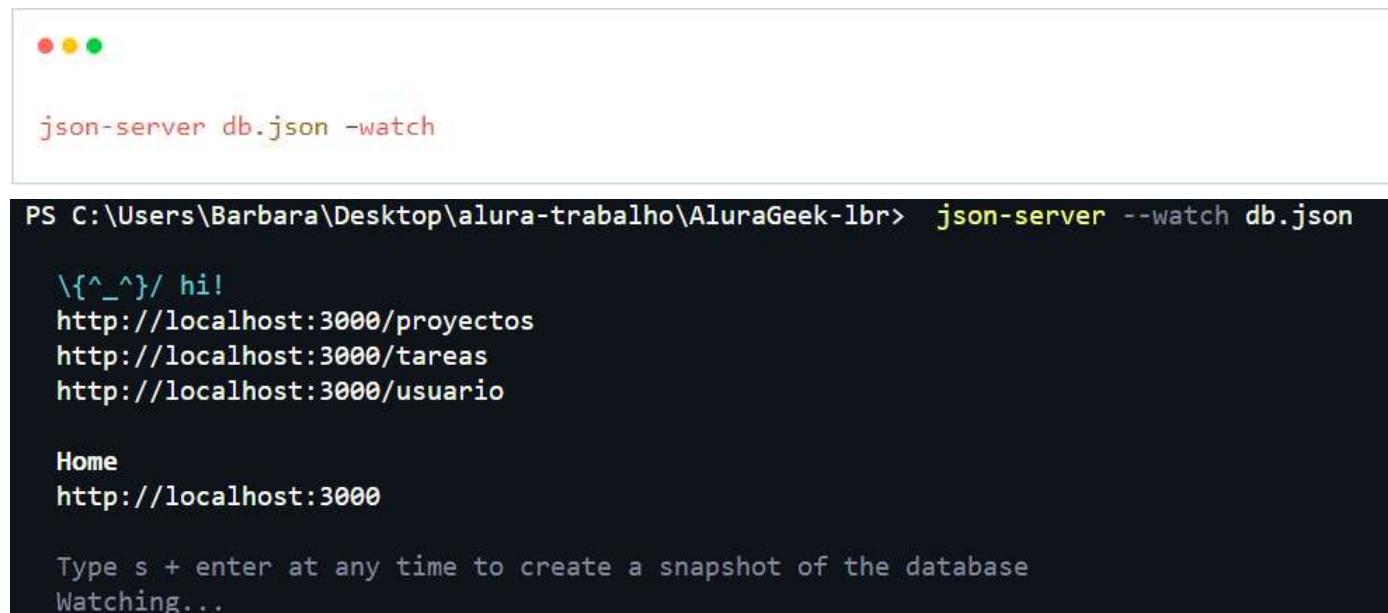
Así que db.json hace que el proyecto funcione. Estas rutas se crean a partir de la estructura de archivos JSON. Entonces, ¿qué sucede cuando tenemos una variedad de

tareas y una variedad de proyectos? Se crean las dos rutas. Incluso podemos ir más allá y crear un endpoint único que devolverá, por ejemplo, los datos del usuario conectado:

```
{
  "proyectos": [
    {
      "id": 1,
      "nombre": "Alura Tracker 3.0"
    },
    {
      "id": 2,
      "nombre": "ByteBank 2.0"
    }
  ],
  "tareas": [
    {
      "id": 1,
      "descripcion": "Configuración del ambiente",
      "proyecto": {
        "id": 1,
        "nombre": "Alura Tracker 3.0"
      }
    },
    {
      "id": 2,
      "descripcion": "Defactorización vuex 4",
      "proyecto": {
        "id": 1,
        "nombre": "Alura Tracker 3.0"
      }
    }
  ],
  "usuario": {
    "id": 13,
    "nome": "Laura González",
    "perfil": "Administración"
  }
}
```

```
}  
}
```

Como cambiamos el db.json manualmente, tenemos que detener el comando y volver a iniciarlo para cargar los nuevos cambios. Siempre que el archivo JSON cambia, idealmente, el servidor json carga automáticamente la nueva información. Así que agreguemos la opción `--watch`:

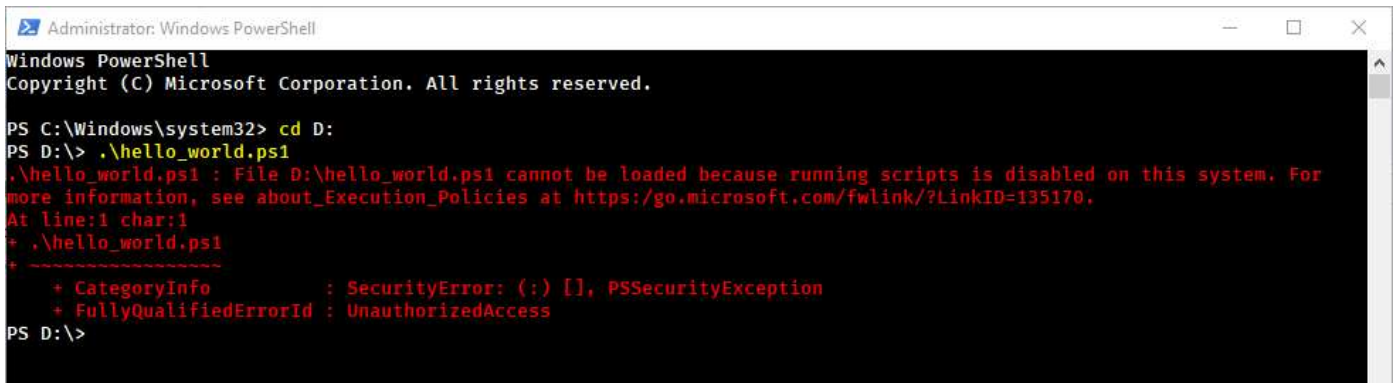


```
json-server db.json --watch  
  
PS C:\Users\Barbara\Desktop\alura-trabalho\AluraGeek-lbr> json-server --watch db.json  
  
\{^_^}/ hi!  
http://localhost:3000/proyectos  
http://localhost:3000/tareas  
http://localhost:3000/usuario  
  
Home  
http://localhost:3000  
  
Type s + enter at any time to create a snapshot of the database  
Watching...
```

Tenga en cuenta que, de acuerdo con la estructura JSON, el servidor json monta los endpoints de las API. Ahora tenemos las tres rutas disponibles y aún mantenemos el servidor json atento a los cambios en el archivo db.json. Ahora, con un excelente backend emulado, ¡estamos listos para avanzar con el desarrollo de frontend hasta que la API real esté disponible!

## No logro instalar el json server, me sale un error, que debo hacer?

Es muy común que cuando uno intente instalar el json server salga un mensaje cómo este:



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd D:
PS D:\> .\hello_world.ps1
.\hello_world.ps1 : File D:\hello_world.ps1 cannot be loaded because running scripts is disabled on this system. For
more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ ~~~~~
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS D:\>

```

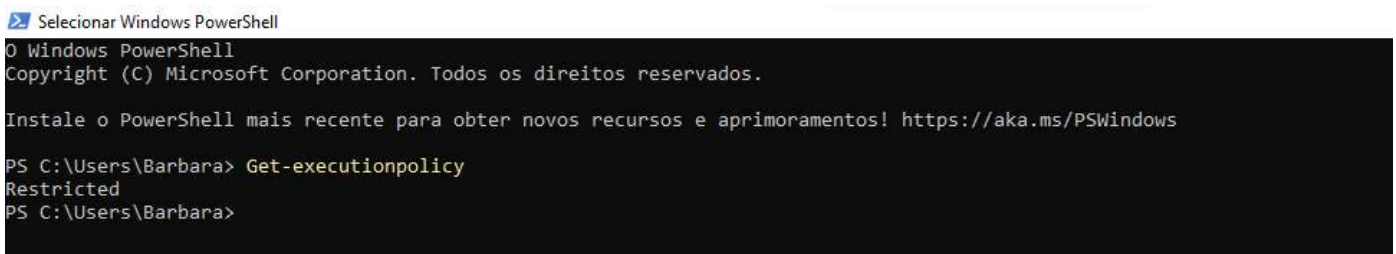
Generalmente cuando este error es exhibido significa que su Windows está con una configuración de políticas de script que no permite instalar determinados paquetes. Para cambiar esta licencia hay que seguir los siguientes pasos:

### 1. Abrir Windows Powershell como administrador



2. Digite el comando `Get-executionpolicy` en el terminal. Es esperado que sea exhibido el valor `Restricted` como en la imagen

### Get-executionpolicy



```

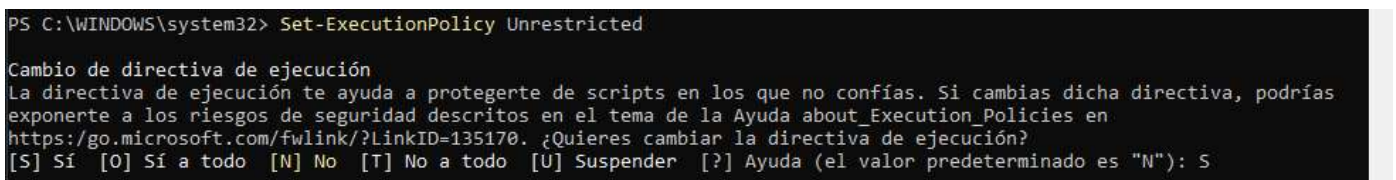
Selecionar Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\Barbara> Get-executionpolicy
Restricted
PS C:\Users\Barbara>

```

3. Si el `Restricted` fue exhibido, digite el comando: `Set-ExecutionPolicy Unrestricted` y el siguiente mensaje aparecerá:



```

PS C:\WINDOWS\system32> Set-ExecutionPolicy Unrestricted

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías
exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): S

```



4. Presione la llave S y la política de restricciones de scripts cambiará , per



```
PS C:\Users\Barbara> Get-executionpolicy
Unrestricted
```

¡Ahora estás listo para empezar tu proyecto!

## Vinicios Neves



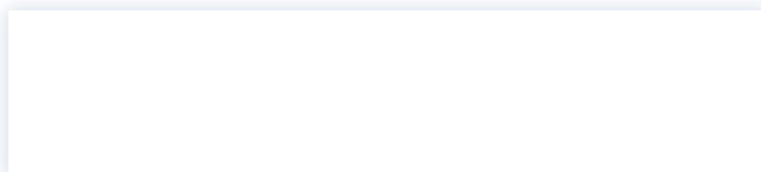
Vinicios es ingeniero de software, involucrado en la arquitectura, diseño e implementación de microservicios, micro frontends y sistemas distribuidos. Tiene una experiencia significativa en aplicaciones, integración y arquitectura empresarial. Es Ingeniero de Software de la UNESA y Arquitecto de Software de la PUC Minas.

traducido y adaptado por **Barbara Santos**

Cursos de Front End

ARTÍCULOS DE TECNOLOGÍA > FRONT END

**En Alura encontrarás variados cursos sobre Front End.  
¡Comienza ahora!**





**SEMESTRAL****US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**[Paga en moneda local en los siguientes países](#)**ANUAL**

# US\$79,90

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

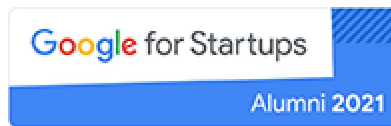
**SÍGUENOS EN NUESTRAS REDES SOCIALES**



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |  
Liderazgo y Gestión de Equipos | Startups y Emprendimiento